

## ПРОЕКТ РЕШЕНИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ С СОЗДАНИЕМ КЛАССОВ МАТЕМАТИЧЕСКИХ ОБЪЕКТОВ

Широкова О.А., к.ф.-м.н., доцент,  
Казанский (Приволжский) федеральный университет, г. Казань  
shirokova2602@mail.ru

*Аннотация.* В статье рассматриваются особенности изучения объектно-ориентированного программирования. При изучении этого курса студентам предлагается создать визуальный проект решения математических задач с созданием классов математических объектов.

*Ключевые слова:* объектно-ориентированное программирование, визуальный проект, модули, классы.

## DRAFT RESOLUTION OF MATHEMATICAL PROBLEMS WITH CREATION OF CLASSES OF MATHEMATICAL OBJECTS

Shirokova O.A.,  
shirokova2602@mail.ru

*Abstract.* In article features of studying of a course of object-oriented programming are considered.. When studying this course students are offered to create the visual project for the solution of mathematical tasks with creation of classes of mathematical objects in Delphi.

*Keywords:* object-oriented programming, visual project, modules, classes.

В современном программировании объектно-ориентированный подход является одним из ведущих. В настоящее время объектно-ориентированный стиль применяется при разработке широкого круга приложений [1-3].

Дисциплина «Современные языки программирования» занимает важное место в системе подготовки студентов математических факультетов. Методическая направленность дисциплины заключается в овладении методом объектно-ориентированного программирования (ООП) и реализации объектно-ориентированной технологии в одной из систем ООП, например, C++ или Delphi. Конечной целью обучения студентов программированию является умение самостоятельно создавать качественные объектно-ориентированные проекты. Практические занятия по дисциплине «Объектно-ориентированное программирование» направлены на формирование навыков разработки таких проектов. Студентов нужно обучать применению знаний в реальных математических задачах, расширяя сферу возможного применения объектно-ориентированного программирования. Для этого рекомендуется решать задачи, имеющие объекты, прототипами которых являются реально существующие математические объекты и структуры.

В статье представлена методика создания проекта реализации класса TProgon в системе объектно-ориентированного программирования Delphi.

Рассмотрим краевую задачу для линейного дифференциального уравнения второго порядка

$$y'' + p(x)y' + q(x)y = F(x), \quad (1)$$

где  $p(x)$ ,  $q(x)$  и  $F(x)$  – заданные функции при  $q(x) < 0$  на отрезке  $[a, b]$ .

Краевые условия заданы в виде:  $y(a) = \alpha$ ,  $y(b) = \beta$ .

Одним из самых эффективных способов решения краевой задачи (1) является сведение ее к системе конечно-разностных уравнений.

$$\begin{aligned} A_i y_{i-1} + B_i y_i + C_i y_i &= f_i; \\ A_i &= 1 - \frac{p_i h}{2}; \\ B_i &= -2 + q_i h^2; \end{aligned} \quad (2)$$

$$C_i = 1 + \frac{p_{ih}}{2}, f_i = h^2 F_i.$$

Система является трехдиагональной и для ее решения используется метод прогонки. Метод прогонки основывается на предположении, что искомые значения функции  $y_i$  связаны рекуррентным соотношением:

$$y_i = \alpha_{i+1}y_{i+1} + \beta_{i+1}, \quad i = n-1, n-2, \dots, 1,$$

которое называется обратной прогонкой.

Здесь коэффициенты  $\alpha_{i+1}$  и  $\beta_{i+1}$  вычисляются методом прямой прогонки

$$\alpha_{i+1} = \frac{C_i}{b_i + a_i \alpha_i}, \quad \beta_{i+1} = \frac{a_i \beta_i - d_i}{b_i - a_i \alpha_i}, \quad i = 0, \dots, n.$$

В ходе прямого хода определяются прогоночные коэффициенты  $\alpha_k$  и  $\beta_k$ . В ходе обратного хода определяются все неизвестные последовательно, начиная с  $y_n$ .

Для разработки визуального проекта решения краевой задачи для линейного дифференциального уравнения второго порядка методом трехдиагональной прогонки создается модуль Progonka с описанием класса TProgon.

Отметим, что длина  $n$  полученных в результате решения массивов  $x_i, y_i$ , не является фиксированной при описании класса TProgon. Длина массивов связана с количеством точек разбиения отрезка и с величиной шага  $h$  сеточных уравнений.

Для вычисления коэффициентов  $A_i, B_i$  и  $C_i$  дифференциального уравнения (2) в классе TProgon описываются методы Show1, Show2, Show3. Здесь действуют соглашения: происходит создание нового файлового документа и полученные результаты выводятся в этот файловый документ. После этого происходит закрытие данного документа.

Отметим, что длина  $n$  полученных в результате решения массивов  $x_i, y_i$ , не является фиксированной при описании класса TProgon. Длина массивов связана с количеством точек разбиения отрезка и с величиной шага  $h$  конечно-разностных уравнений.

Фрагмент описания модуля Progonka, в котором описан класс TProgon с методами, необходимыми для решения краевых задач представлен ниже:

```
Unit Progonka;
interface
uses SysUtils;
type TProgon=class
a, b, h: real;
n: integer;
constructor Create(New_a, New_b: real; New_n: integer);
function p(x: real) : real;
function q(x: real) : real;
function Ai(x: real) : real;
function Bi(x: real) : real;
function Ci(x: real) : real;
function DM1(x: real) : real;
function f1(x: real) : real;
function DM2(x: real) : real;
function f2(x: real) : real;
function DM3(x: real) : real;
function f3(x: real) : real;
procedure Show1(x:array of real; y:array of real);
procedure Show2(x:array of real; y:array of real);
procedure Show3(x:array of real; y:array of real);
end;
implementation
function TProgon.p(x: real): real;
begin
```

```
Result := 0;
end;
```

Разработка визуального проекта начинается с моделирования и структурирования, то есть создания проекта системы. В мире программного обеспечения для этого служат модели UML. Модель UML – это абстракция, описывающая суть сложной проблемы или структуры без акцента на несущественных деталях, тем самым делая ее более понятной. Модели помогают организовывать, отображать, понимать и создавать сложные проекты.

При моделировании необходимо создать диаграмму классов. Между классами TProgon и TForm1 существует отношение композиции.

Представим диаграмму классов объектно-ориентированного проекта, предназначенного для решения задач с применением классов TProgon и TForm1 (рис. 1).

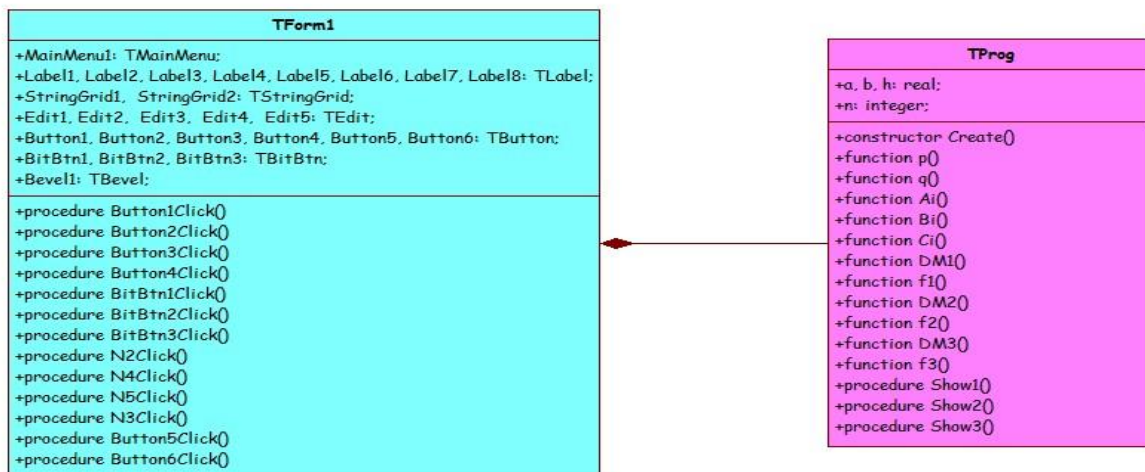


Рис. 1 Диаграмма классов приложения с применением класса TProgon.

После создания UML-модели проекта нам необходимо визуально разработать интерфейс главного окна проекта. Интерфейс проекта создается с помощью меню, размещенного на форме (рис. 2). Меню диалогового окна позволяет эффективно решать краевые задачи для линейного дифференциального уравнения (1):

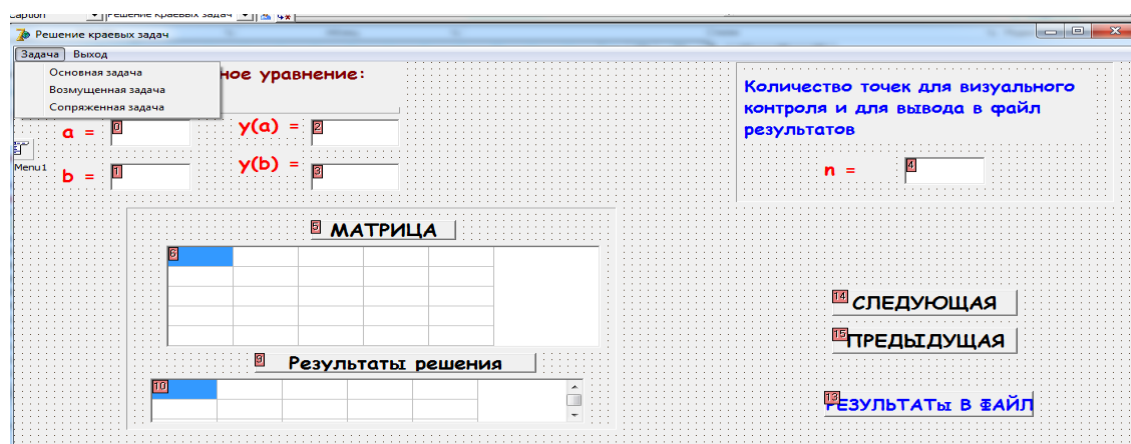


Рис. 2 Расположение компонентов на форме.

В визуальном проекте предусматривается вывод результатов приближенного решения методом прогонки рассматриваемой краевой задачи, а также вывод трехдиагональной матрицы соответствующей системы конечно-разностных уравнений.

Обучение студентов объектно-ориентированному стилю программирования способствует формированию умений и навыков по формализации задачи, выделению абстракций и объектов

данной предметной области, по моделированию и реализации их с помощью объектно-ориентированной технологии программирования.

### **Литература**

1. Широкова О.А. Особенности обучения программированию на основе общности и различия принципов // Современные проблемы науки и образования. – 2015. – № 1, с.1757. URL: <http://www.science-education.ru/121-17896>
2. Широкова О.А. Объектно-ориентированные проекты решения математических задач //Материалы XI Международной науч.-практ. Конф. «Объектные системы – 2015» (Ростов н/Д: ШИ (ф) ЮРГПУ (НПИ) им. М.И.Платова, 2015. – С.15-23.
3. Gainutdinova T.U., Shirokova O.A. Features of Professional Teachers Training of Informatics in a Programming Course / T.U.Gainutdinova, O.A.Shirokova //The European Proceedings of Social & Behavioural Sciences EpSBS. – 2016. – Pp. 31-37.